**SCHEMA LOGICO**

UTENTE(IdUtente, Username, Password, Email, Premium (booleano))

BLOG(IdBlog, Titolo, Descrizione, Immagine, IdUtente\*, IdCategoria\*)

COAUTORE(IdUtente\*, IdBlog\*)

CATEGORIA(IdCategoria, Nome)

CONTIENE(IdSopracategoria\*, IdSottocategoria\*)

POST(IdPost, Titolo, Data, Ora, Testo, Immagine, IdBlog\*, IdUtente\*)

COMMENTA(IdUtente\*, IdPost\*, Data, Contenuto)

FEEDBACK(IdUtente\*, IdPost\*, Tipo)

CREATE ASSERTION Scrittura

CHECK (NOT EXISTS

(SELECT IdUtente FROM POST

WHERE (POST.IdUtente NOT IN (SELECT COAUTORE.IdUtente FROM COAUTORE WHERE POST.IdBlog = COAUTORE.IdBlog))

OR (POST.IdUtente NOT IN (SELECT BLOG.IdUtente FROM BLOG WHERE POST.IdBlog = BLOG.IdBlog))))

*L’assertion* ***Scrittura*** *impone che, per scrivere un post, un utente debba essere necessariamente o un co-autore di blog o un creatore*

CREATE VIEW NumeroBlog(IdUtente, NumeroBlog) AS

SELECT IdUtente, COUNT(IdBlog)

FROM BLOG

GROUPBY IdUtente

WITH CHECK OPTION

*La view* ***NumeroBlog*** *visualizza il numero totale di blog creati da ciascun utente*

CREATE ASSERTION NumeroBlog

CHECK (3 >= ALL(SELECT COUNT(BLOG.IdBlog)

FROM BLOG, UTENTE

WHERE BLOG.IdUtente = UTENTE.IdUtente AND UTENTE.Premium = 0 GROUPBY BLOG.IdUtente))

*L’assertion* ***NumeroBlogStandard*** *impone che un utente standard possa creare fino a un massimo di 3 blog*

CREATE ASSERTION Sottocategoria

CHECK (1 >= SELECT COUNT(IdSottocategoria)

FROM CONTIENE

GROUPBY IdSottocategoria)

*L’assertion* ***Sottocategoria*** *impone che ciascuna sottocategoria afferisca al più ad una sopracategoria*

**DATABASE**

CREATE TABLE Utente(

IdUtente SMALLINT AUTO\_INCREMENT PRIMARY KEY,

Username VARCHAR(25) NOT NULL,

Passw VARCHAR(30) NOT NULL,

Email VARCHAR(30) NOT NULL,

Premium BIT NOT NULL

); CREATE TABLE Categoria(

IdCategoria SMALLINT AUTO\_INCREMENT PRIMARY KEY,

Nome VARCHAR(25) NOT NULL

); CREATE TABLE Blog(

IdBlog SMALLINT AUTO\_INCREMENT PRIMARY KEY,

Titolo VARCHAR(30) NOT NULL,

Descrizione TEXT,

Immagine BLOB,

IdUtente SMALLINT REFERENCES Utente(IdUtente) ON DELETE CASCADE ON UPDATE CASCADE,

IdCategoria SMALLINT REFERENCES Categoria(IdCategoria) ON DELETE SET NULL ON UPDATE CASCADE

); CREATE TABLE Coautore(

IdUtente SMALLINT REFERENCES Utente(IdUtente) ON DELETE CASCADE ON UPDATE CASCADE,

IdBlog SMALLINT REFERENCES Blog(IdBlog) ON DELETE CASCADE ON UPDATE CASCADE,

PRIMARY KEY(IdUtente, IdBlog)

); CREATE TABLE Contiene(

IdSopracategoria SMALLINT REFERENCES Categoria(IdCategoria) ON DELETE CASCADE ON UPDATE CASCADE,

IdSottocategoria SMALLINT REFERENCES Categoria(IdCategoria) ON DELETE CASCADE ON UPDATE CASCADE,

PRIMARY KEY(

IdSottocategoria,

IdSopracategoria

)); CREATE TABLE Post(

IdPost SMALLINT AUTO\_INCREMENT PRIMARY KEY,

Titolo VARCHAR(30),

Data DATE NOT NULL,

Ora TIME NOT NULL,

Testo TEXT,

Immagine BLOB,

IdBlog SMALLINT REFERENCES Blog(IdBlog) ON DELETE CASCADE ON UPDATE CASCADE,

IdUtente SMALLINT REFERENCES Utente(IdUtente) ON DELETE CASCADE ON UPDATE CASCADE

); CREATE TABLE Commenta(

IdUtente SMALLINT REFERENCES Utente(IdUtente) ON DELETE SET NULL ON UPDATE CASCADE,

IdPost SMALLINT REFERENCES Post(IdPost) ON DELETE CASCADE ON UPDATE CASCADE,

PRIMARY KEY(IdUtente, IdPost),

Data DATE NOT NULL,

Contenuto TEXT NOT NULL

); CREATE TABLE Feedback(

IdUtente SMALLINT REFERENCES Utente(IdUtente) ON DELETE SET NULL ON UPDATE CASCADE,

IdPost SMALLINT REFERENCES Post(IdPost) ON DELETE CASCADE ON UPDATE CASCADE,

PRIMARY KEY(IdUtente, IdPost),

Tipo BIT NOT NULL

);